	Type	Hits	Search Text	DBs
1	BRS	2	"6678880".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
2	BRS	2	"5119475".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
3	BRS	53	"Rational Rose" and "class diagram"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
4	BRS	38	S3 and "multiple inheritance"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
5	BRS	38	S4 and pointer	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
6	BRS	38	S5 and handler	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
7	BRS	38	S6 and (user near interface)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
8	BRS	38	S7 and drag	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB

	Type	Hits	Search Text	DBs
9	BRS	38	S8 and drop	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
10	BRS	36	S9 and hierarchy	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
11	BRS	4	"Rational Rose/C++"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
12	BRS	164	"Rational Rose"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
13	BRS	5	S12 and (rose near model)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
14	BRS	10	("object oriented" near "modeling tool")	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
15	BRS	11	("object oriented" near (modeling near tool))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
16	BRS	26	(rational near2 tool) and "object oriented"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB

	Type	Hits	Search Text	DBs
17	BRS	16	"5119475".pn. or "5339433".pn. or "5410648".pn. or "5418964".pn. or "5432903".pn. or "5455599".pn. or "5493680".pn. or "5499371".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB
18	BRS	22	"5537630".pn. or "5546519".pn. or "5560014".pn. or "5572733".pn. or "5642511".pn. or "5644771".pn. or "5692195".pn. or "5699310".pn. or "5761511".pn. or "5832487".pn. or "6070006".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB

P⊚RTAL

US Patent & Trademark Office

Search: ⦿ The ACM Digital Library   ○ The Guide

+"visual object-oriented programming"   **SEARCH**

■ Feedback Report a problem Satisfaction survey

Published before May 1995
Terms used **visual object oriented programming**

Found 6 of 67,222

| Sort results by | relevance ▣ |
| Display results | expanded form ▣ |

● Save results to a Binder
? Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 6 of 6

Relevance scale ☐▢▬▬■

1 Visual object-oriented programming   ■

Margaret M. Burnett
April 1993 **ACM SIGPLAN OOPS Messenger , Addendum to the proceedings on Object-oriented programming systems, languages, and applications (Addendum),** Volume 5 Issue 2

Full text available: ▣ pdf(425.63 KB)   Additional Information: full citation, index terms

2 Joyce: an object-oriented decision tree builder   ▢

B. Marcus
May 1989 **ACM SIGPLAN Notices**, Volume 24 Issue 5

Full text available: ▣ pdf(264.86 KB)   Additional Information: full citation, abstract, index terms

Joyce is an object-oriented visual programming environment for nonprogrammers. It permits an application developer to assemble a program by graphically choosing methods from a large library and linking them together using predefined control structures. Joyce also provides the developer with the ability to define data structures, interactive forms, and documentation without programming.

3 The value of a baseline in determining design success   ▢

Brenda Burkhart, Darold Hemphill, Scott Jones
April 1994 **Conference companion on Human factors in computing systems**

Full text available: ▣ pdf(69.46 KB)   Additional Information: full citation

4 Recursive interfaces for reactive objects   ▢

Michael Travers
April 1994 **Conference companion on Human factors in computing systems**

Full text available: ▣ pdf(69.46 KB)   Additional Information: full citation

5 LiveWorld: a construction kit for animate systems   ▢

Michael Travers
April 1994

Conference companion on Human factors in computing systems

Full text available: pdf(710.51 KB)   Additional Information: full citation

6 Recursive interfaces for reactive objects

Michael Travers

April 1994 **Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence**

Full text available: pdf(1.03 MB)   Additional Information: full citation, references, citings, index terms

Keywords: agents, direct manipulation, objects, programming environments, rules, visual object-oriented programming

Results 1 - 6 of 6

PⓇRTAL

US Patent & Trademark Office

Subscribe (Full Service)   Register (Limited Service, Free)   Login

Search: ◉ The ACM Digital Library   ○ The Guide

+"visual object-oriented programming"   [SEARCH]

Feedback Report a problem Satisfaction survey

Published before May 1995
Terms used **visual object oriented programming**

Found **6** of **67,222**

Sort results by   [relevance ▾]
Display results   [expanded form ▾]

📘 Save results to a Binder
❓ Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 6 of 6

Relevance scale ☐ ▭ ▬ ▬ ■

■

1 Visual object-oriented programming
Margaret M. Burnett
April 1993 **ACM SIGPLAN OOPS Messenger , Addendum to the proceedings on Object-oriented programming systems, languages, and applications (Addendum),**
Volume 5 Issue 2
Full text available: 📄 pdf(425.63 KB)   Additional Information: full citation, index terms

▭

2 Joyce: an object-oriented decision tree builder
B. Marcus
May 1989 **ACM SIGPLAN Notices,** Volume 24 Issue 5
Full text available: 📄 pdf(264.86 KB)   Additional Information: full citation, abstract, index terms

> Joyce is an object-oriented visual programming environment for nonprogrammers. It permits an application developer to assemble a program by graphically choosing methods from a large library and linking them together using predefined control structures. Joyce also provides the developer with the ability to define data structures, interactive forms, and documentation without programming.

▭

3 The value of a baseline in determining design success
Brenda Burkhart, Darold Hemphill, Scott Jones
April 1994 **Conference companion on Human factors in computing systems**
Full text available: 📄 pdf(69.46 KB)   Additional Information: full citation

▭

4 Recursive interfaces for reactive objects
Michael Travers
April 1994 **Conference companion on Human factors in computing systems**
Full text available: 📄 pdf(69.46 KB)   Additional Information: full citation

▭

5 LiveWorld: a construction kit for animate systems
Michael Travers
April 1994

Conference companion on Human factors in computing systems

Full text available: 📄 pdf(710.51 KB)   Additional Information: full citation

6 Recursive interfaces for reactive objects

Michael Travers

April 1994 **Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence**

Full text available: 📄 pdf(1.03 MB)   Additional Information: full citation, references, citings, index terms

Keywords: agents, direct manipulation, objects, programming environments, rules, visual object-oriented programming

Results 1 - 6 of 6

h   c   g e   cf

P⊕RTAL

US Patent & Trademark Office

Search: ⦿ The ACM Digital Library   ○ The Guide

+"Rational Rose"

SEARCH

Feedback Report a problem Satisfaction survey

Published before May 1995
Terms used **Rational Rose**

Found **3 of 67,222**

| Sort results by | relevance ▣ |
| Display results | expanded form ▣ |

❧ Save results to a Binder
🄿 Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 3 of 3

Relevance scale ☐ ☐ ☐ ☐ ☐

1 Generating ECAD framework code from abstract models   ☐
Joachim Altmeyer, Bernd Schürmann, Martin Schütze
January 1995 **Proceedings of the 32nd ACM/IEEE conference on Design automation**

Full text available: 🄰 pdf(60.96 KB)   Additional Information: full citation, references, citings, index terms

2 An object-oriented, distributed architecture for large-scale Ada systems   ☐
Phillipe Kruchten, Christopher J. Thompson
November 1994 **Proceedings of the conference on TRI-Ada '94**

Full text available: 🄰 pdf(1.14 MB)   Additional Information: full citation, abstract, references, citings, index terms

This paper presents an architectural model ideally suited for the description of large, distributed command and control systems. This model is organized around multiple dimensions (or views) of software architecture and is used to describe the software architecture of a family of automated air traffic control systems currently under development by Hughes Aircraft of Canada. Some of the features of this family of systems are described, and in particular the mechanism used for transparent acc ...

3 Preliminary defect data from the iterative development of a large C++ program (experience report)   ☐
James F. Walsh
October 1992 **ACM SIGPLAN Notices , conference proceedings on Object-oriented programming systems, languages, and applications**, Volume 27 Issue 10
Full text available: 🄰 pdf(644.66 KB)   Additional Information: full citation, references, citings, index terms

Results 1 - 3 of 3

Useful downloads: 🄰 Adobe Acrobat   🄰 QuickTime   🄰 Windows Media Player   🄰 Real Player

h   c   g e   cf

Visual Object-Oriented Programming

Report by:

Margaret M. Burnett
Oregon State University

Attendance at the Workshop:

Organizers:

Margaret Burnett
Adele Goldberg
Ted Lewis

Attendees:

Marjan Bace
Ed Baroth
Bay-Wei Chang
S.-K. Chang
Wayne Citrin
Phil Cox
John Hosking
T. Dan Kimura
Dan LaLiberte
Chris Laffra
David Leibs
Stephen MacKay
John Maloney
David McIntyre
Toshimi Minoura
Francesmary Modugno
Wolfgang Pree
Albert Schappert
Stefan Schiffer
Randy Smith
Michael Travers
David Ungar
Russel Winder.

Aim of the Workshop

More and more researchers and developers are recognizing the potential of visual programming to make OOP easier and less error-prone. Yet, in spite of much recent activity in the area of visual programming, there are still a number of unsolved problems. The purposes of this workshop were both to determine and to advance the state of the art for visual OOP by:

1. Presenting and discussing recent and emerging work in the area of visual object-oriented programming

2. Discussing unsolved problems and directions for possible solutions

3. Generating new ideas via an active exchange among researchers and developers in the area

Format of the Workshop

The workshop consisted of three sessions, followed by a wrap-up and an informal demonstration period. The first session focused on the general ideas behind visual object-oriented programming, and especially the challenges and opportunities. The second was about the use of visual techniques to achieve both power and simplicity in object-oriented programming. The third was centered on open issues and unsolved problems. In each session, there were three or four topics covered in some depth, each led by a presenter or discussion leader. The presentations were short and usually highly interactive, with the attendees contributing comments and discussion throughout the presentation. Some of the presenters also had live systems or videos to demonstrate their work.

Session 1: Visual Object-Oriented Programming: What's the Point?

One of the main outcomes of the workshop was an increased awareness of what visual object-oriented programming is really trying to accomplish, or perhaps even more importantly, the things that it should be trying to accomplish. Adele Goldberg began the first session with an introduction and subsequent discussion into this question. She made the points that we should think more about what it is about visual techniques that could (1) help us express ourselves to the computer, and (2) help ourselves and others to understand the program once it has been written. She also pointed out a number of challenges, including scalability, debugging, reusability, and some interesting questions regarding appropriate representations. The notion of appropriateness generated a lot of discussion throughout the workshop. Phil Cox's presentation raised eight common objections to the concept of visual programming and pointed out the fallacies of these objections. Finally, David Leibs discussed the benefits of visual techniques versus the benefits of text, and talked about how these benefits can be integrated to work together. His presentation was done using a live

system to illustrate his points with examples from VisualWorks.

During this session, a number of observations were made during discussion. Among them were:

- "Visual" does not always mean "non-textual." Several examples were given of visual arrangements of text.

- Careful attention as to the intended audience is needed, both when discussing the advantages and disadvantages of some particular approach and when designing a visual object-oriented programming system. It was not felt that there is necessarily any one particular audience for visual programming in general, but that there usually is for some particular visual system.

- Rightly or wrongly, syntax is often viewed as relatively unimportant for textual languages. However, this is unacceptable in visual languages.

- One issue that is important for visual OOP systems is harmony between the language and the environment.

Session 2: Power and Simplicity

This session was about using visual techniques to achieve power, but to do so more simply than has been done textually. Bay-Wei Chang presented work he is doing jointly with David Ungar, Randy Smith, and John Maloney on a visual programming environment for the prototype-based object-oriented language Self. The goal of this work is to focus exclusively on objects in a very concrete way, so that the user will regard the image of the object on the screen is the actual object. They have incorporated several techniques originally inspired by cartoonists' use of animation to reduce cognitive load in helping the user avoid figuring out what is changing and how. The discussion centered on whether multiple views would be appropriate for such a concrete approach, and how abstractions might be employed for scalability without giving up the concreteness.

David McIntyre presented a visual programming system called VAMPIRE for creating visual programming languages. The system is rule-based, featuring before and after images for specification of these languages. Dan Kimura demonstrated his pen-based computing system, which is being integrated with a visual object-oriented system called Hyperflow. The system is still being developed. One of his goals is to further explore the relationship between visual languages and user interfaces. His term for the research is "visual software engineering."

Session 3: Probing the Future: Issues and Unsolved Problems

Margaret Burnett started the third session by pointing out a number of issues in visual object-oriented programming, and the ensuing discussion added to this list. Some of these issues were:

- So far, visual environments for textual OOP languages have had more success overall than visual OOP languages. One way to look at the difference is by describing it as an issue of granularity. Is there a granularity at which visual

representations lose their effectiveness by nature, or is it simply a matter of time before appropriate representations for low-level programming details become further developed and accepted?

- Visual OOP is relatively new, and little attention has been paid so far to software engineering in this approach. What are the software engineering challenges and opportunities for visual OOP?

- Is visual OOP usable by real programmers trying to get complex, sophisticated programs written and debugged?

- There are a number of programming-language issues that are critical to visual OOP languages. Among them are the need for suitable approaches to polymorphism, to inheritance or delegation, and to the issue of sequence and state modification in a visual world.

- Visual OOP system designers need to be more cognizant of what the strengths of visual programming actually are, and to build upon those strengths.

Some of these issues were expanded upon by the ensuing presenters. S.-K. Chang's presentation was about the issue of dealing with complexity. He presented a visual specification system for modeling and prototyping real-time distributed systems. The idea is to integrate software engineering methods with Petri Nets as a way to achieve formal support in a visual approach. Stefan Schiffer led a discussion on the issues of software engineering for visual programming. He made the point that visual programming researchers need to pay more attention to software engineering issues, and that OOP with its many software engineering advantages may be just the ingredient needed to allow this to happen. Wolfgang Pree contributed the idea of visual techniques to support software engineering from a structural point of view.

Russel Winder continued the software engineering theme in leading his discussion of the impact of the granularity issue on software engineering concerns. Finally, Ed Baroth concluded the session back in the "real world" by a factual presentation of the use of visual programming at NASA JPL for real, production-programming tasks. His experience has shown a reduction in software development time by a factor of 4 to 10. He also reports higher user satisfaction with the resulting programs. He described the approach to program development in his organization, and the ensuing discussion included much speculation as to exactly which aspects of the visual programming effort were leading to the most improvement.

Wrap-Up Session

Ted Lewis wrapped up the workshop by distilling how the issues that were being raised in the discussions and presentations fit into a bigger picture. He pointed out several categories of issues that had been raised. There were:

- The question of the goals of visual OOP: Is it for end-user programming, professional programmer productivity, or something else?

- More attention to the appropriateness of various visual techniques that are used in visual OOP: He made an analogy to different kinds of books, and pointed out that for some kinds of books an almost purely textual form is most appropriate (e.g., a dictionary), and that for other kinds of books an almost purely pictorial form is most appropriate (e.g., a photo album).

- There are several difficult issues specific to the OOP part of visual OOP that have hardly been explored at all.

- The question of sequencing in a visual programming environment is difficult and needs more attention.

- The issues of data, data mappings, and representations of data are still relatively unstudied in visual OOP systems.

- The problem of scalability is perhaps the most pressing of all, and includes many of the issues that were raised during the workshop.

Outcomes

There was a general agreement that the workshop had achieved the goals set forth in the Call for Participation, which are restated in the first section of this report. Many of the participants felt they were taking away at least one new idea that they could use in some way in their work. A more concrete outcome of the workshop is to be a book on Visual Object-Oriented Programming, containing full versions of many of the abstracts prepared for this workshop by the participants. This will allow the workshop participants to share with a wider audience what the workshop collectively found to be the current state of visual OOP, as well as descriptions of recent and emerging work in the area.